

# UML Diagram

**Before:**

<b>GeneratePassword</b>
- password : string - userId : string
+ GeneratePassword() + readFile() - writeToFile(string, string) - createRandomPasswords() : string

<b>Encryption</b>
- key : string - keyArray: char
+ Encryption() + readFile() - writeToFile(string, string) - encryptPasswords(string) : string

<b>HashTable</b>
- hashTable : HashTable[ ] - head : HashTable* - next : HashTable* - TABLE_SIZE : const int - data : string
+ HashTable() + readFile() + search(string) - createTable() - hash(string) : int - insertIntoTable(string, string)

<b>Testing</b>
ROW : const int COLUMN : const int testArray : string
+ Testing() + readFile() + testingLegalPasswords(string, string) + testingIllegalPasswords(string, string)

**After:**

<b>GeneratePassword</b>
- inFS : ifstream - outFS : ofstream:
+ GeneratePassword(string, string) - readFile(string, string) - writeFile(string, string) - createRandomPasswords() : string

<b>Encryption</b>
- key : string - keyArray: char - inFS : ifstream - outFS : ofstream
+ Encryption(string, string) - readFile(string, string) - writeFile(string, string) - encryptPasswords(string) : string - generateKeyArray()

<b>HashTable</b>
- **hashTable : HashNode[ ] - TABLE_SIZE : const int - inFS : ifstream - outFS : ofstream
+ HashTable(string, string) - readFile(string, string) + searchHashTable(string) - createHashTable() - hash(string) : int - insertIntoTable(string, string) + getEncryptedPassword(string) : string - writeHashTableToFile()

<b>HashNode</b>
+ data : string + keyUserId : string + password : string + *next : HashNode
+ HashNode(string, string, string)

+ HashNode
------------

TestProgram
<ul style="list-style-type: none"><li>- inFS : ifstream</li><li>- outFS : ofstream</li><li>- ROW : static const int</li><li>- COLUMN : static const int</li><li>- testArray : string</li></ul>
<ul style="list-style-type: none"><li>+ TestProgram(string, string)</li><li>+ testingBothCases(Hash, Encryption)</li><li>- readFile(string, string)</li><li>- changeOneCharInPassword(string) : string</li><li>- testingLegalPasswords(Hash, Encryption)</li><li>- testingIllegalPasswords(Hash, Encryption)</li></ul>